(12) UK Patent Application (19) GB (11) 2 311 389 (13) A

(43) Date of A Publication 24.09.1997

(21) Application No 9605669.2

(22) Date of Filing 18.03.1996

(71) Applicant(s)
International Business Machines Corporation

(Incorporated in USA - New York)

Armonk, New York 10504, United States of America

(72) Inventor(s)
Roger Philip Hoggarth
Richard Ian Knox
Andrew Liam Massey
Colin David McCall
Sohail Syyed

(74) Agent and/or Address for Service
D P Litherland
IBM United Kingdom Limited, Intellectual Property Department, Mail Point 110, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(51) INT CL^6
G06F 9/445

(52) UK CL (Edition O)
G4A AFL

(56) Documents Cited
GB 2255877 A **US 5421009 A**

(58) Field of Search
UK CL (Edition O) **G4A**
INT CL^6 **G06F 9/445 11/30**
Online: WPI, INSPEC, COMPUTER

(54) Software installation in data processing network.

(57) A client is arranged so that at power up it makes a request to a server for first boot code which allows the client to boot from its own mass storage. Since the OS is not sent from the server over the network, traffic flow is eased whilst strict control of client software type is maintained.

The configuration of a client may be checked prior to software, particularly OS, installation by a server. The server, detecting a new client on the network, down-loads scan code onto the client. The client then runs the scan code and returns hardware configuration data to the server.

If the OS of the client is to be modified then the server sends different, second boot code to the client which instructs the client to boot (a new OS) directly from the server. The OS from the server is then installed on the client mass storage device, and subsequent IPLs take place using the first boot code.

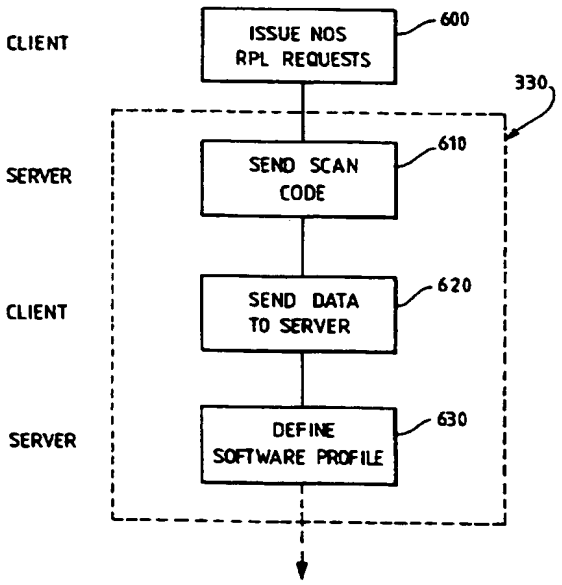

FIG. 6

GB 2 311 389 A

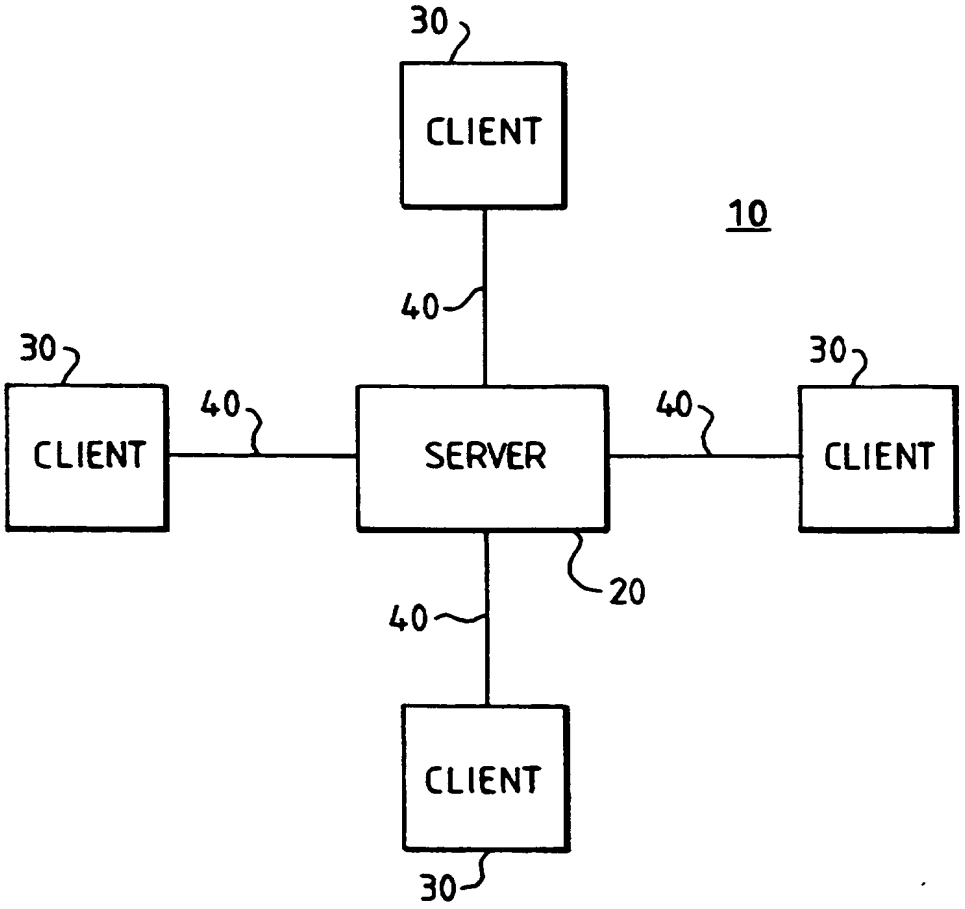

FIG. 1

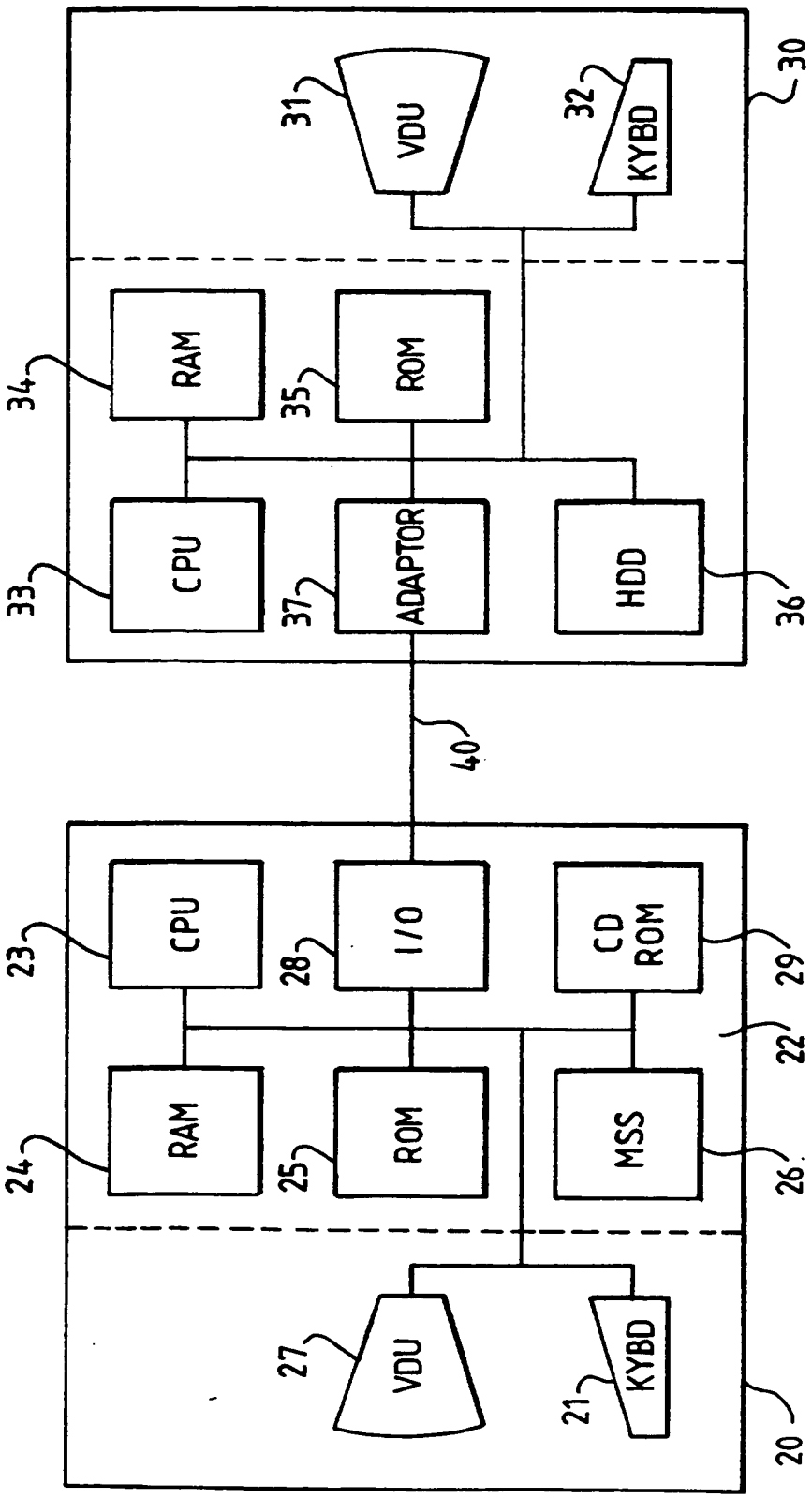

FIG. 2

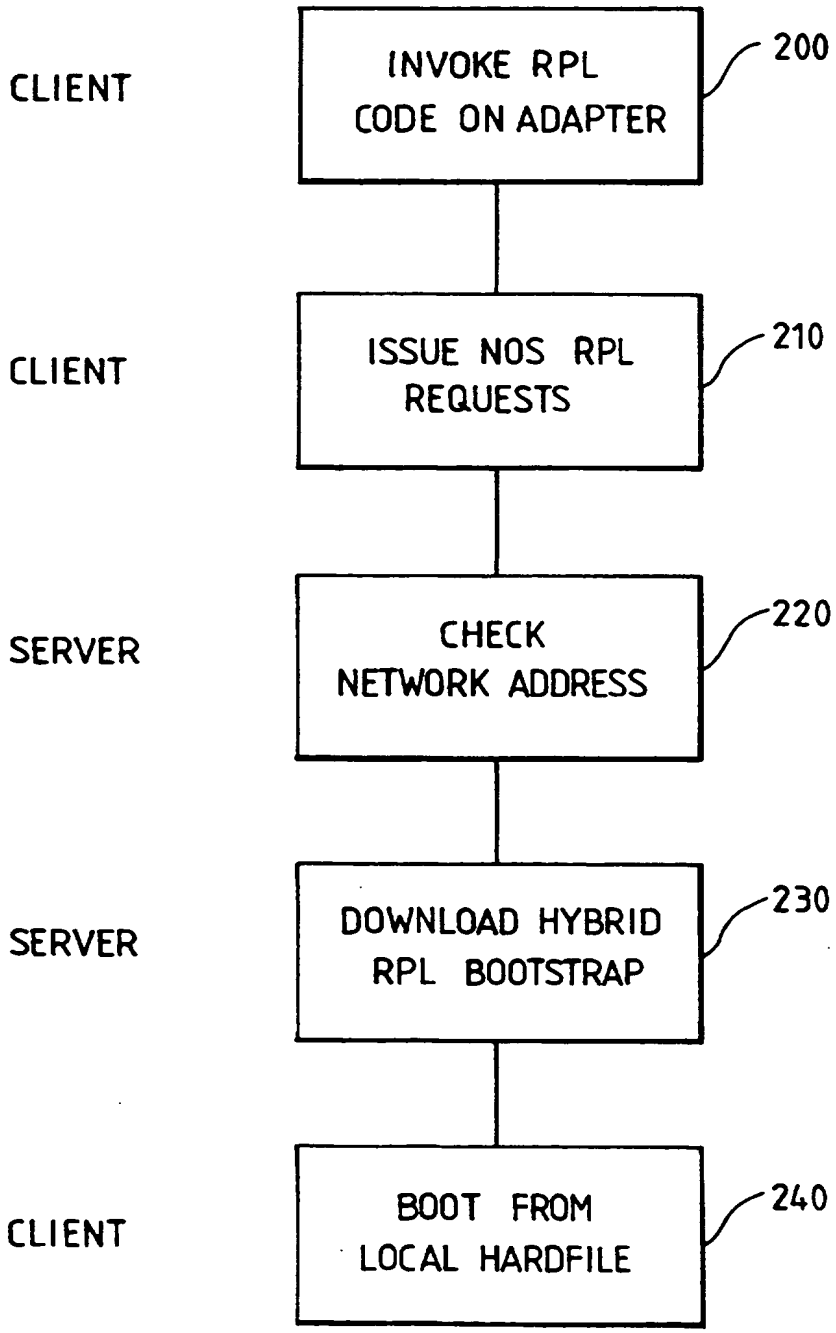

FIG. 3

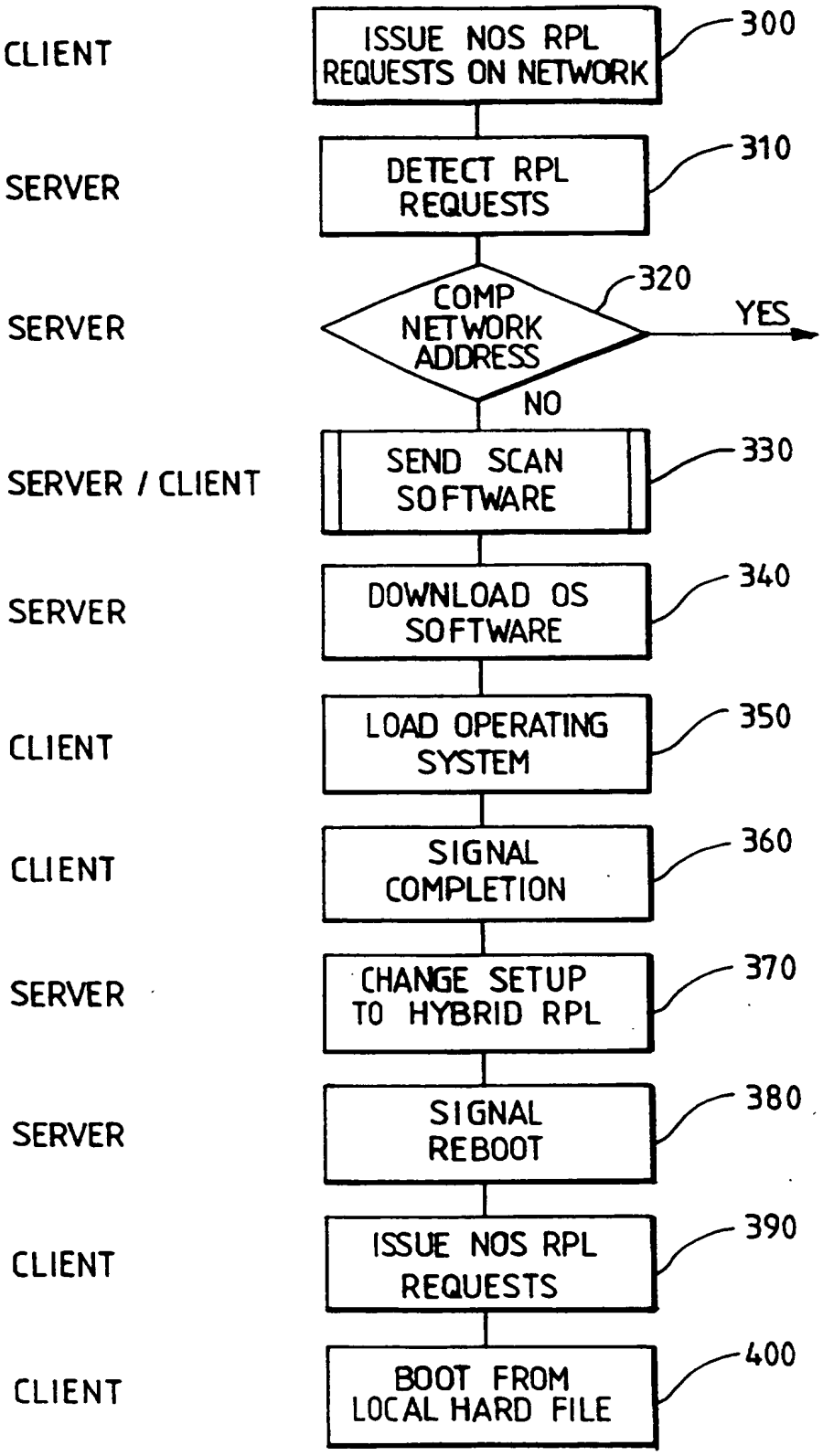

FIG. 4

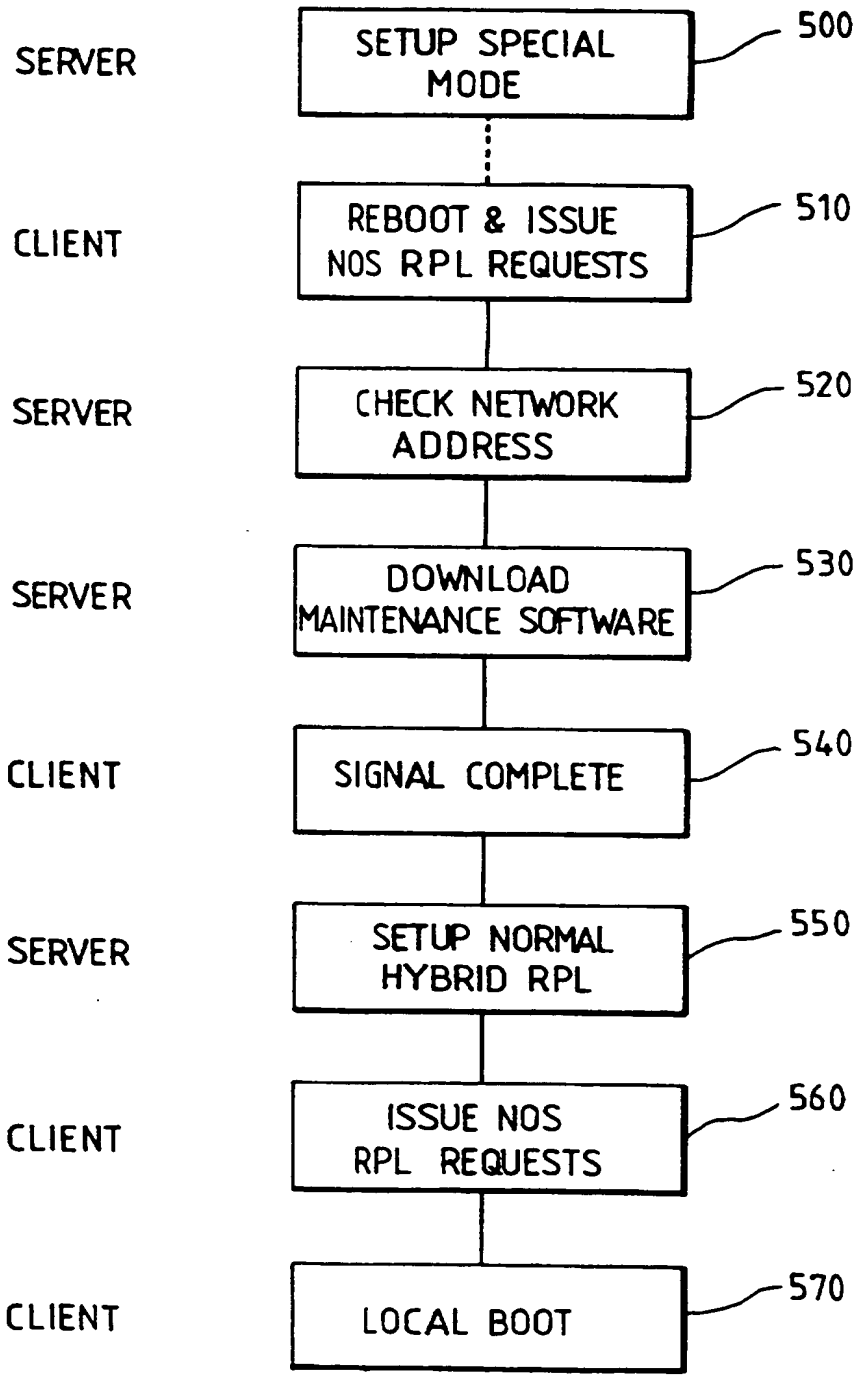

FIG. 5

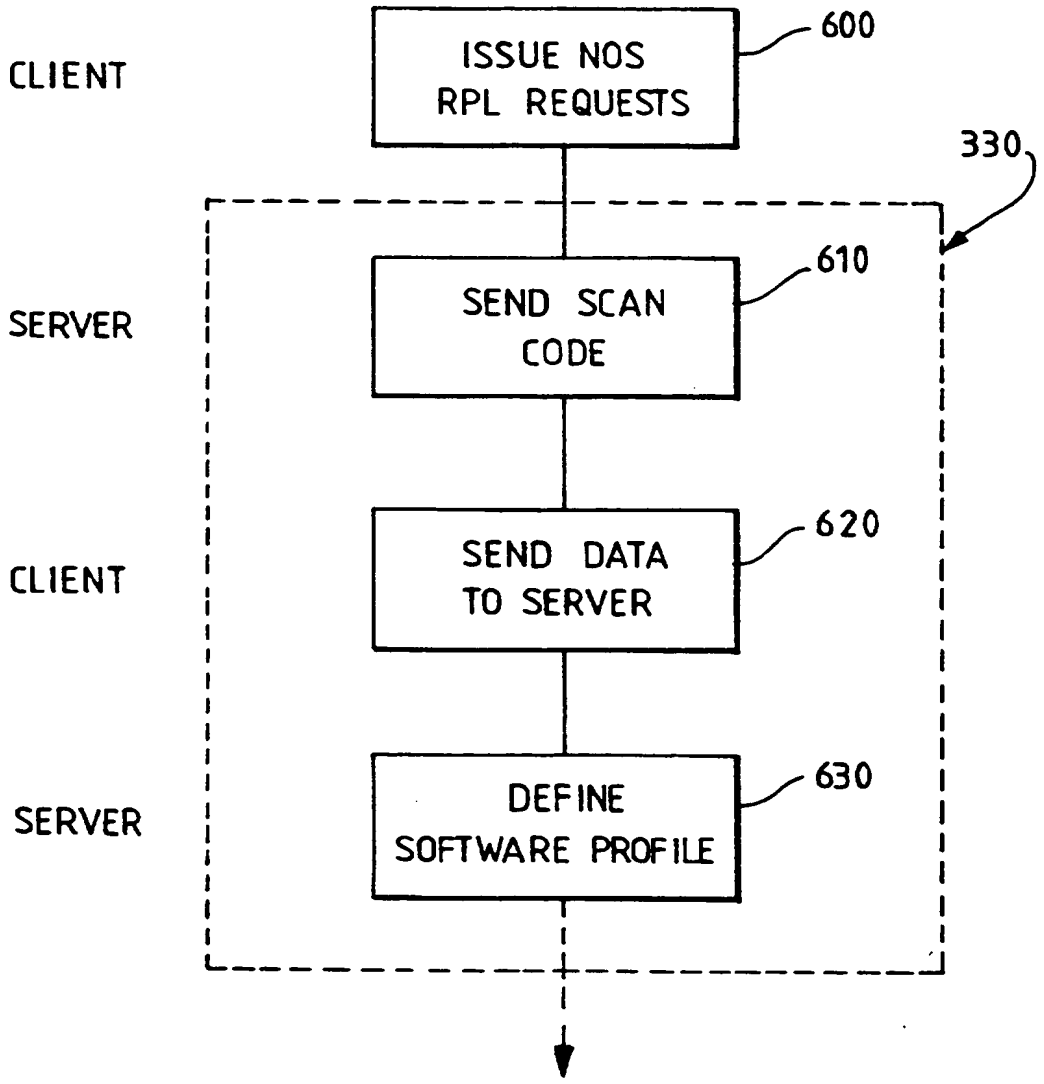

FIG. 6

SOFTWARE INSTALLATION IN DATA PROCESSING NETWORK

Technical Field of the Invention

The present invention relates to a data processing network of the type in which a plurality of client computer systems are connected to a server computer system. More particularly, the invention relates to the installation of software on client computer systems in such a network.

Background of the Invention

In a typical network environment, multiple client computer systems (clients) are connected to one or more server computer systems (servers). In a first common arrangement, each client system includes an operating system and optionally other software, stored on a hard file within the client. On power-up or reboot, the client boots from the operating system stored on the hard file without reference to the server computer. This type of boot is called a local program load. Other application software e.g. word processing, database software etc, held on storage associated with the server system, is accessed as needed by the client system.

It is currently common for software to be installed on the hard file of a client system at the factory as part of the system manufacturing process i.e. operating system and application software is preloaded onto the hard file after the system has been assembled. Alternatively, the software is sold with the computer system recorded on floppy disks or CD-ROMs. The system user himself then has to carry out what is often a lengthy and complicated process to install the software onto the system hard file.

In the computer network environment, various techniques are known whereby operating system and application system software can be downloaded onto a client computer from a server computer. One such technique is disclosed in US patent 5142680 in which a subset of the operating system software is loaded into the memory of the computer which is to receive the operating system. The subset of the operating system contains the basic commands for file creation and manipulation and network communication. The computer system is then started using the subset of the operating system located in memory and connected to the network. Once the computer is connected to the network, the files comprising the operating system to be downloaded are copied and

transferred from remote computer over the network and stored on the disk drive of the receiving computer.

Another technique is the so-called custom JumpStart installation for automatically installing the SOLARIS operating system onto a network. Before installation, a suite of programs called 'sysidtool' is used to configure the target system. The configuration is done either by requiring the user of the target system to answer a series of questions before the software is installed, or alternatively, by providing default values for sysidtool which may be set automatically. Although this technique appears to go some way to automating the configuration component of the automation process, it is not clear that it is able to achieve automation of the configuration process for a network having a variety of client systems with differing hardware configurations and consequently differing device driver and other software requirements.

Accordingly, it would be desirable to provide an improved technique for automating the configuration component of the software installation process.

Disclosure of the Invention

Accordingly, in a first aspect of the present invention, there is provided a method of configuring a client computer system attached to a server computer system in a computer network comprising: downloading configuration determination program code from the server computer system to the client computer system over the network; executing the program code on the client system to gather data on the hardware configuration of the client system; and transferring the gathered configuration data to the server computer system.

In a preferred method, the step of downloading is executed on receipt of a remote initial program load request from the client computer. This initial program load request is generated by the client computer when first connected to the network and powered-on.

Automated software installation is achieved according to a preferred method whereby in response to the receipt of the gathered configuration data, the method further comprises the steps of determining the software profile for the client system based on the configuration data, and downloading operating system software, as defined by the

determined profile, to the client system for storage on a mass storage device thereof.

According to a second aspect of the invention, there is provided a data processing network comprising a server computer system connected for communication to a plurality of client computer systems, said server system including means for downloading program code to a target client system for determining the hardware configuration of said client, said client being responsive to receipt of said code to run said code to gather data on the client hardware configuration and to transfer the gathered configuration data to said server.

In a preferred data processing system, the server system is responsive to the receipt of the configuration data to determine the software profile for the client system based on the configuration data and to download operating system software, as defined by the determined profile, to the client system for storage on a mass storage device thereof.

In one network arrangement, the server computer system comprises a main server and a centralised control computer (similar to a network client system) which is used by the network administrator to configure and control the network. In such an arrangement, the gathered configuration data is transferred from the client to the control computer which causes the server to download the operating system software to the client.

According to a further aspect of the present invention there is provided a server computer system which includes means for transferring the scan code to a target client system and is responsive to receipt of configuration data received from the client system to download, to the client, operating system software according to a software profile which is dependent on the configuration data.

By means of the present invention, the hardware configuration of a client system can be readily determined without the need for client-user intervention. Furthermore, in contrast with prior art techniques, the scan software is designed to execute on the target client system without the need for any operating system software already being stored on the client system. However in certain embodiments, it may be advantageous to send a portion of the operating system code with the scan code to enable

more efficient scanning. This operating system code is downloaded into system RAM to enable the scan program code to use standard system facilities for network communication and optionally for communication with the client system user. The configuration data gathered by the scan program code is automatically sent to the server.

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings.

Brief Description of the Drawings

Figure 1 is a schematic representation of a computer network according to a preferred embodiment of the present invention;

Figure 2 is a block-diagrammatical representation of a client computer system connected to a server computer system in the network of Figure 1;

Figure 3 is a flowchart showing the normal program load procedure;

Figure 4 is a flowchart showing the program load procedure during software installation;

Figure 5 is a flowchart showing the program load procedure during client maintenance;

Figure 6 is a flowchart showing details of the scan process according to a preferred embodiment of the present invention to determine the hardware configuration of a client system prior to software installation.

Detailed Description of the Invention

A preferred embodiment of the scanning process of the present will next be described in the overall context of an automated software installation process. While not necessary for the purposes of describing the present invention, the reader is referred to applicant's copending application (applicant's reference number UK9-96-020) for further details of the improved user interface provided for automatic software installation.

Referring firstly to Figure 1, there is shown, in schematic form, a local area network (LAN) 10 according to one embodiment of the invention. The network of Figure 1, which may be constituted as an Ethernet or Token-ring LAN or other arrangement, is constituted of a server computer system 20 (which may be an IBM PC 700 computer system) connected for communication by link 40 with a plurality of client computer systems 30. The client computer systems may be personal computers based on the Intel X86 family of microprocessors or other form of computer system. Typically such personal computers include a LAN adapter card to provide communication with the server computer. Control of resources on the network including communication between server and clients is effected by means of a network operating system (NOS) e.g. OS/2 LAN Server from IBM Corporation having a 'server' component which executes on the main processor(s) of the server computer system and a corresponding 'requester' component which executes on the main processor of each client computer system. Other suitable network operating systems include Netware from Novell Inc and OS/2 WARP server from IBM.

Figure 2 is a simplified block diagram in which a server computer system 20 is shown connected to a client system 30 over a communication link 40. The client system, constituted in the present embodiment by a personal computer, includes a keyboard 31 and a display 32 operating under the control of control logic in the form of main CPU 33 which is connected by a system bus to system memory (RAM) 34 and non-volatile memory (ROM) 35, in which is stored system BIOS. The control logic is connected to one or more mass storage devices 36 e.g. in the form of a magnetic disk drive or hard file or similar. If it is desired to prevent the client user from introducing software or data into the client system, the client system is advantageously provided without a diskette drive, CD-ROM drive or similar device which would otherwise allow the user to introduce data and/or software via a floppy disk or CD-ROM. As already mentioned, the client system further includes a network adapter card which, in the present embodiment, may be either an ethernet or token-ring adapter card. This adapter card provides the communication between the client and server. In an alternative arrangement, the network attachment function provided by the adapter card is alternatively provided by control logic on the main client system motherboard.

The server computer system, which is configured to have more processing power and mass storage than a client system, includes a keyboard 21 attached to a system unit 22 including a main CPU 23, system

RAM 24, system ROM 25 and mass storage capability 26, typically in the form of multiple magnetic disk drives constituted in a RAID (redundant array of independent disks) arrangement. Stored on the server mass storage devices are a variety of different types of software including different operating system software e.g. Windows 95, OS/2 WARP, Windows/DOS etc and application software, copies of which are selected for storage on the hard file of each client system. The server system may optionally include a display 27 (if the network administrator requires direct interaction with the server system -- the network administrator may otherwise use a separate console system (not shown in the figures) similar in hardware configuration to the already described client systems) and other storage devices such as a diskette drive (not shown) and CD-ROM drive 29. Input/output logic 28 provides communication with the client system. The network administrator, using a console system, will be provided with certain privileges, not available to a client user, which allows him to control the network.

Although not indicated in Figures 1 or 2, the network may additionally comprise a further one or more server systems connected to a subset or all of the client computer systems. Furthermore, the client systems may not all be of the type described above. For example, the network may include so-called 'medialess' workstations i.e. systems which do not include a mass storage device and which are thus incapable of storing a local operating system. Such systems are configured to RPL from the server system at boot time in a conventional manner.

As discussed above in the background section, client systems in existing networks either boot from a server by means of a remote program load (RPL) operation, or alternatively execute a local program load from an operating system stored locally on the hard file. As will be described below in more detail, there is provided a 'hybrid' initial program load mechanism which combines the advantages of both conventional RPL and local program load techniques while avoiding at least some of their disadvantages.

In essence the hybrid remote/local program Load (Hybrid RPL) technique operates in two modes: Normal and Special.

In the Normal case, at power-on or reboot of a client system on which is installed a local operating system, the client issues requests for RPL on the network. As will be described in greater detail below, the

client system BIOS is configured such the client may only boot from the server and not from a local hardfile or diskette. In response to the client RPL request, the server sends the client a bootstrap program which initiates a local program load which causes the client to operate, in effect, like a normal (i.e. non-RPL) client system.

In the Special case, when the network administrator wishes to take control of the workstation e.g. for maintenance purposes, the administrator firstly changes the setup of the server system to specify a different RPL bootstrap program for that client e.g. a minimal operating system with a remote maintenance utility. At the next power-on or re-boot, the client system issues a request for RPL as usual. The server then responds with the special bootstrap program as defined by the administrator. In response to the bootstrap program, the client then operates like a conventional RPL client, loading the software specified by the administrator. However, unlike most conventional RPL clients where the software is loaded for operation into the client's volatile memory, the software is instead installed onto the client hard file. The network administrator then changes the setup of the server to specify a local program load for the client. At the next power-on or reboot, the client then reverts to the normal operation as described above and carries out a local program load from the 'amended' software.

The hybrid RPL technique is preferably implemented as software code executing on the server computer, providing a user interface to the network administrator to allow him to configure and control the network in the manner to be described below.

It is an important feature of the hybrid RPL technique that each client is 'forced' to send a boot request from the server at each power-on or reboot. That is, even where the client has a local operating system stored on its hard file, the client issues an RPL request to the server at each power-on or reboot. Each client system which is operable under the hybrid RPL technique must include some non-volatile storage device such as a hard file or similar, on which a portion of, or complete operating system may be stored to enable the normal mode hybrid RPL operation. The client is prevented from taking control and carrying out an unsupervised local boot. In the present embodiment, it is the system BIOS in the client which specifies the location from where the client boots at power-on or reboot. Thus the system BIOS is set to specify that

the client may boot only from the network. This restriction is set into the system BIOS at the factory.

It will be appreciated that while setting of the system BIOS will generally ensure that the client can only boot from the network, it is possible that the client user could, if so-minded, gain access to, and reset the client system BIOS to specify a local boot. It is therefore desirable to employ a disabling mechanism, in the form of a BIOS password, by which the client user can be prevented from tampering with the system BIOS settings, including the location of the boot device. BIOS passwords are employed in some existing personal computers e.g. IBM PC 700 to prevent unauthorised access to the BIOS setup routines. Thus the unauthorised user is prevented from resetting details of the device from which the client system boots. Although BIOS passwords per se are already known, one advantage of the present technique is that the BIOS password is settable remotely by the network administrator. The BIOS password may be set by the network administrator when the client is first installed in the network. Alternatively, the password may be set in the factory at the time the client system is manufactured. One drawback with this latter technique is that some mechanism would then be required to convey the BIOS password to the server when the client is first installed in the network.

A different BIOS password may be set for all or a subset of clients. Alternatively, a single BIOS password may be employed for all clients. As will be described below, the BIOS password for selected clients may be reset at will by the network administrator by invocation of the 'special' mode of the Hybrid RPL mechanism.

Referring now to Figures 3, 4 and 5, the detailed operation of both normal and special modes will be described. As has been mentioned, the method steps are preferably implemented as a central process provided on the server computer which controls the operation of the client system.

Figure 3 is a flowchart showing the sequence of steps involved in a normal case hybrid RPL operation. This sequence of steps is carried out when the client system is initially powered-up e.g. when the client user turns on the machine when wishing to start work, or alternatively at reboot. It is assumed that the client system already has a local operating system installed on its hardfile from which it can boot.

At step 200, after the client has completed various tests invoked at power-on, the system BIOS causes the client to invoke RPL code stored on the network adapter card. At step 210, this RPL code, executing on the main processor of the client, causes the client to issue NOS RPL requests over the link connecting the client to the server. A central process executing on the server, which listens for RPL requests from attached clients recognises, at step 220, the client unique network address in the request and at step 230 sends hybrid bootstrap code to the client. As indicated at step 240, the hybrid bootstrap program causes the client to boot from its local hard file. The hybrid bootstrap loads the master boot record from the hard file to a fixed location in RAM, checks the content to ensure it is a valid master boot record and passes program control to it.

Once the boot process is complete, the client user is then able to operate the client in a largely conventional manner. Thus, in principle, the client operates with no load on the network except for the transfer of the initial RPL requests and hybrid RPL bootstrap code.

In a preferred arrangement however, the client system is configured by the network administrator such that the local hard-file is read-only. That is, the client user is able to read files from the hardfile but may only access and store read/write files or data files on the server system. This feature is enabled, for example, by providing drivers for the hard file during software installation which restrict end-user access to the client hard file. In general, the drivers are advantageously designed to block all system calls which (i) create files; (ii) delete files; (iii) open files for writing; and (iv) change file attributes. For example for client systems loaded with DOS/Windows, access to the hard file is restricted by intercepting DOS software interrupts for file handle operations. This hard file access protection feature provides even greater control over the contents of a user's system at the cost of some increase in network loading. It will be appreciated however, that this loading will be much lower than in conventional RPL networks.

Next will be described the operation of the special mode of hybrid RPL. As will be apparent, the special mode can be invoked by the network administrator to handle a variety of different types of situations. In a first situation, the special case is used to provide for the automatic installation of operating system and optionally other administrator-specified software when the client system is first connected to the

network. Alternatively, the special case mode can be invoked to provide for re-installation or upgrade of the software already stored on the client hard file. In a further alternative, the special case mode can be invoked to reflash the system BIOS or reset the BIOS password. Other maintenance actions can be carried out in this manner.

With reference to Figure 4, there will now be described the sequence of steps involved in providing automatic installation of software onto the client system when the system is first connected to the network. One advantage of providing for automatic installation of software in this way is that it removes the need to carry out time-consuming and equipment-intensive preloading operations at the factory. A particular advantage of the present technique is that the network administrator has complete control over the installation and no client-user intervention is required except for physically connecting the client to the network and flipping the power switch. It will be further appreciated that some network adapters allow clients to be powered-on remotely, in which case there will be no need to the client user to power-on the system. This latter ability would permit software installation to take place during time of low network traffic (i.e. during the night).

According to the present invention, a scanning technique is employed such that on initial installation, the hardware configuration of the client is established by means of scan code issued by the server which executes on the client system, which in turn sends the results back to the server. The server then selects the appropriate software profile (either automatically or through the intervention of the network administrator) for the target client system based on its hardware capabilities (e.g. depending on amount of memory available, addressability of graphics adapter etc) and decisions made by the network administrator. In this way is enabled the automatic assignment of software to the client systems based on their hardware characteristics.

Before installation into the network, the client system hardfile will have no installed operating system. The system BIOS (which specifies that the client can boot only from the network) is already installed in the system. The network adapter card is connected via cabling attached to its external connector to the server.

At step 300 in Figure 4, the client is powered-on which causes the client network adapter card to issue NOS RPL requests over the network. The NOS RPL requests specify the network address which was burned-in to the client at the factory. At step 310, a central process executing on the server system detects the RPL requests issued by the client. At step 320, the server process compares the network address of the client against a locally-stored list of known clients. As the client is new to the network, the new client address will not be in the list and the server therefore marks the target client as requiring scanning. Details of the scan process are shown and described below with reference to Figure 6, but briefly the server process sends the scan client software to the client at step 330 which determines the hardware configuration, details of which are then sent to the server. Based on this information, the server assigns a software profile to the client and at step 340 downloads, to the client, an install program and operating system and optionally other software according to the client profile stored in the server. At step 350, the client receives the code from the server and runs the install program to load the operating system software onto the client hardfile. At step 360, the client indicates completion of the installation process by sending a complete signal to the server. The server responds, as indicated at step 370, by changing the setup for the client to normal mode of hybrid RPL.

At step 380, the server issues a signal to the client causing the client to reboot. The client reboots, at step 390, and issues NOS RPL requests to the server. In response, the server transmits the hybrid bootstrap code (as described above in relation to step 230 of Figure 3) to the client which in turn causes, at step 400, the client to boot from the newly-installed operating system software from its hard file. The client may then continue operation.

As is described in more detail in copending application (applicant's reference number UK9-96-020), the server system is provided with software which allows the network administrator to create profiles for each connected client. These profiles include the software images that are to be installed onto the client systems and optionally further information tailored to a specific client. For example, a network may be comprised of multiple clients having identical hardware configurations. In this case, it may be suitable for the administrator to define, via the server software, a common software image for each client. However, it will more commonly happen that the network will be comprised of client

systems having differing hardware configurations, in which case it is advantageous to tailor the software images as appropriate. This tailoring is also referred to as personalisation.

The aim of the installation process is to fully install as wide a range of software as possible onto the hard files of the clients. In a preferred arrangement, the installation stage indicated at step 340 of Figure 4 comprises three main elements.

a) Depending on the particular software profile, selected clients may need preparation to receive the final software image. Some operating system software (e.g. Windows 95) does not load onto standard FAT partitions on the hardfile. In this case, pre-processing is required to prepare suitable disk partitions. A pre-load image is provided to the client which specifies an initial set of software to be executed on the client before the final software image is loaded.

b) The final image is copied onto the client's hardfile. The image file is advantageously a self-extracting .exe file that contains all the files required by the client. This is copied down to each client and is then expanded onto the clients' local hard file. By this means, each client has stored thereon an identical software image on their hard files.

It is however generally necessary to personalise each client for a variety of reasons. For instance, most networking systems require a unique TAG for each client on the network e.g. TCP/IP addresses may need to be set to allow a protocol stack to operate; terminal emulation programs may require different id's for each client. This personalisation is achieved by the provision of personality files associated with each client. These are run as the last step in the installation (or re-installation) process.

c) The client personality files defined by the network administrator are downloaded and run on each client. These files are simple batch files that allow .INI, text and registration database files to be edited for each client.

Next will be described, with reference to Figure 5, the sequence of steps involved in the special hybrid RPL mode during maintenance operations. As will be appreciated, a large number of these steps are the

same as steps described above in relation to Figure 4. As has been described, the special mode is invoked by the network administrator when it is desired to take control over the client for re-installation or other maintenance purposes.

At step 500, the network administrator changes the setup of the server to specify a new hybrid RPL profile for a client. In other words, the 'normal' mode of operation is disabled. At step 510, when the client system is next powered-on or rebooted (either using a remote control utility specified by the administrator or alternatively by the client user), the client issues one ore more NOS RPL requests via the network adapter. At step 520, the server recognises the client unique network address in the request. However, in this mode, the server does not issue the hybrid bootstrap code to cause the client to execute a local boot. Instead, the server downloads selected software according to a software profile defined according to the required maintenance function. In the following description, this downloaded software is termed maintenance software though as will be described below it may in fact be software for upgrading the local operating system, software for upgrading system BIOS or other software. At step 530 therefore, the maintenance software is downloaded to the client system where it is executed and/or stored onto the local hardfile. At step 540, the client indicates to the server that the operation on the client is complete. In response, the server process changes, at step 550, the setup for the client back to hybrid RPL bootstrap. At next reboot, indicated at step 560, which may be initiated either by the remote control utility or by the client user, the client issues the NOS RPL request via the client network adapter card and local boot takes place as per normal hybrid RPL - step 570.

In one example, the maintenance program comprises an operating system upgrade; which may be either a complete operating system or additional upgrade code to add to the existing software. Thus, the server downloads a revised operating system and install program which replaces the operating system currently stored on the hardfile. Alternatively, the maintenance software can be an upgraded version of the client system BIOS which is re-written over the existing BIOS in flash ROM while preserving any client specific information such as serial number and model number. In this latter case, the server advantageously invokes an immediate reboot of the client after the upgraded BIOS is loaded in order to make use of the new level of system BIOS.

In a further arrangement, the maintenance software comprises code for resetting the password associated with the system BIOS. The password is stored in non-volatile storage on the client in the form of an encoded check sum and an indication that the password function is enabled. The password is changed by writing a different checksum to the non-volatile RAM, or disabled by writing an indication that the password function is disabled.

Details of the initial scan process will now be described. As has been described above, this network scan technique for clients gathers details of any new clients which are attached to the network. The scan process is loaded to each client as it initiates RPL. With reference to Figure 6, when the client is first attached to the network, it issues one or more NOS RPL requests onto the network, at step 600, which identify the network address of the client. As the client is new to the network, the network address specified in the request does not exist on the client address list held in the server and accordingly, the central server process marks the received address as requiring an RPL of the scan client software. Therefore at step 610, the server responds by sending the scan client software to the client which then executes on the client to gather information about the client hardware configuration. The scan code executes on the client system processor. Information is gathered by one or more of :

(i) BIOS function calls which return the information;
(ii) Reading system identification information directly from the hardware on the client system board, including the system processor itself;
(iii) Reading information contained in the BIOS ROM;
(iv) Reading information written by the BIOS ROM into the non-volatile storage on the system;
(v) Where the client system bus allows it, reading identification information from adapter cards in the client system bus. (Microchannel and PCI busses, among others, enable this function;
(vi) Optionally, by prompting the user of the client system to type in additional information, for example the location of the system or the telephone number of the user.

For example, the information gathered includes (i) network adapter type, (ii) network address, (iii) hard disk presence and storage capacity, (iv) RAM installed, (v) graphics adapter type. Optionally, the

information gathered further includes (i) client serial number, (ii) client machine type and model, (iii) whether there is a mouse attached, (iv) whether there is a keyboard attached, (v) any additional user defined data.

Although the scan process is designed to avoid client user intervention, the network administrator may also define additional information to be collected at install time, such as the location or contact name. This is done by specifying a prompt file which causes prompts to be displayed at the client during the scan process. Responses typed by the client user are recorded.

At step 620, the client sends the gathered configuration data to the server, where the information is assimilated and a software profile appropriate to the target client is defined -- step 630. The software installation process continues as already described.

It will be appreciated that a number of advantages over existing techniques are provided by the hybrid RPL technique described above. In particular:

The network administrator retains full control over each client's software. At any time the RPL setting can be changed to special mode for any client so that the next time the client boots, new software can be downloaded onto that client.

The network administrator retains full control over client operations. If the client is removed from the network and an attempt is made to reboot, the RPL will fail as there is no network to respond to the client systems RPL requests. The BIOS boot setting is protected by password and therefore it is not possible for the client user to boot from the local hardfile.

CLAIMS

1. A method of configuring a client computer system attached to a server computer system in a computer network comprising

downloading configuration determination program code from the server computer system to the client computer system over the network;

executing the program code on the client system to gather data on the hardware configuration of the client system; and

transferring the gathered configuration data to the server computer system.

2. A method as claimed in claim 1, wherein the step of downloading is executed on receipt of an initial program load request from the client computer.

3. A method as claimed in claim 1 or claim 2 wherein in response to the receipt of the gathered configuration data, the method further comprises the steps of determining the software profile for the client system based on the configuration data, and downloading operating system software, as defined by the determined profile, to the client system for storage on a mass storage device thereof.

4. A data processing network comprising a server computer system connected for communication to a plurality of client computer systems, said server system including means for downloading program code to a target client system for determining the hardware configuration of said client, said client being responsive to receipt of said code to run said code to gather data on the client hardware configuration and to transfer the gathered configuration data to said server.

5. A data processing network as claimed in claim 4, wherein said server system is responsive to the receipt of the configuration data to determine the software profile for the client system based on the configuration data and to download operating system software, as defined by the determined profile, to the client system for storage on a mass storage device thereof.

6. A data processing network as claimed in claim 4 or claim 5, wherein said server computer system comprises a server and a centralised control computer, said gathered configuration data being transferred from said client to said control computer.

7. A data processing network as claimed in claim 6 as dependent on claim 5, wherein, in response to the receipt of the gathered configuration data at the control computer, said control computer causes the server to download the operating system software to the client.

The Patent Office

Application No:	GB 9605669.2	**Examiner:**	Matthew Gillard
Claims searched:	All	**Date of search:**	2 May 1996

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A.

Int Cl (Ed.6): G06F 9/445, 11/30.

Other: On-line: WPI, INSPEC, COMPUTER.

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	GB 2255877 A	(DOWTY). See page 8 and the figure.	1
A	US 5421009	(HEWLETT-PACKARD). See column 6, lines 19-26.	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

This Page Blank (uspto)